Software Managed Networks
Portfolio Task – C-Lab-Report - 3

Leo Holden
Dept. Of Computing Technologies
Swinburne University of Technology
Hawthorn, Australia
103996982@student.swin.edu.au

I. Aim

The aim of this lab report is to demonstrate practical experience and use of Graphic Network Simulartor-3 (GNS3), for creating, configuring and testing a virtual network. Configurations will be completed through basic GNS3 setup, Open vSwitch (OVS) and Ubuntu host template set up, assigning IP addresses to interfaces and establishing internet connectivity. Various commands will be issued to verify configurations and correct functionality. Furthermore, configurations are completed to ensure that the statically assigned IP addresses remain even after device restarts.

II. Equipment

The virtualized environment is produced through VMware creating a Linux virtual machine (VM) preinstalled with relevant tools. Running a 64-bit Ubuntu version 18.04.5 LTS, the VM has been allocated the following hardware resources: 8 GB RAM, 6 cores and a 40 GB disk partition, the VM has GNS3 version 2.2.22, the OVS used in GNS3 has Open vSwitch version 2.17.12 supporting OpenFlow version 1.0-1.5. *Table 1* shows the specification of the local machine and VM. The VM should be capable of operating at minimal hardware allocations e.g. 2 GB of RAM and 2 cores.

Specification	Local Machine	Virtual Machine
CPU	12th Gen Intel i5-12500H (12 cores / 16 threads)	6 virtual CPUs allocated
RAM	16 GB	8 GB
Storage	512 GB	40 GB
Operating System	Windows 11 64-bit	Ubuntu 18.04.5 LTS 64-bit
Virtualisation Tool	VMware Workstation 17	
Software Versions		GNS3 2.2.22, Open vSwitch 2.17.12, OpenFlow 1.0-1.5

Table 1 Local hardware and virtual machine's specifications

III. Method

As per lab handout: P-Lab-06-GNS3

IV. Results

This lab is segmented into five sections that outlines what is performed which are sections: *A. Import GNS3 Appliances, B. GNS3 Appliance Configuration, C. Connect GNS3 Devices to SDN-VM, D. Establish Internet Connectivity, E. Static IP Persistence and DNS Configuration.*

A. Import GNS3 Appliances

GNS3 has extensive library of preconfigured devices which can be pulled from the GNS3 online marketplace repository [2], the GNS3 documentation can be found in [1]. To import the OVS switch, navigation to the left side appliance sidebar and clicking the `browse all devices`, this shows the current available appliances, Click: `New Template`.

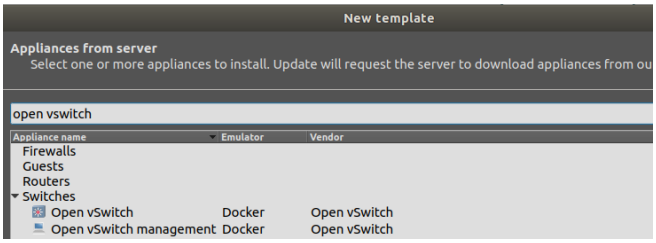

Figure 1 Import Open vSwitch template

Accepting default selections and entering Open vSwitch in the search bar will show the appliances. The procedure can be carried out to obtain the Ubuntu Docker Guest appliance.

B. GNS3 Appliance Configuration

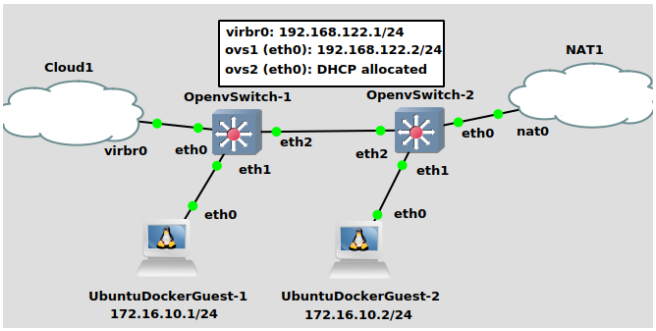

Figure 2 GNS3 Network topology 1

Links can be added through the left side bar too and connected according to *Figure 2*. Every device can be

activated through clicking the large green play button in the top menu bar. Red squares indicate down interfaces and green dots indicate the interface is up. To prevent layer 2 loops, Spanning Tree Protocol (STP) needs to be enabled to prevent excessive CPU usage. The following commands are used: `ovs-vsctl set bridge br0 stp_enable=true`, to verify issue; `ovs-vsctl get bridge br0 stp_enable`.

```
root@UbuntuDockerGuest-1:~# ifconfig eth0 172.16.10.1 netmask 255.255.255.0
root@UbuntuDockerGuest-1:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 0a:0b:dd:eb:d0:d6
          inet addr:172.16.10.1  Bcast:172.16.10.255  Mask:255.255.255.0
          inet6 addr: fe80::80b:ddff:feeb:d0d6/64 Scope:Link
```

Figure 3 Ubuntu guest eth0 IP config

Figure 3 shows how to configure and verify an interface IP address configuration for the Ubuntu guest-1 appliance. The configuration is repeated for guest-2 with the respective IP address outlined in *Figure 2.*

```
root@UbuntuDockerGuest-2:~# ping 172.16.10.1
PING 172.16.10.1 (172.16.10.1) 56(84) bytes of data.
64 bytes from 172.16.10.1: icmp_seq=1 ttl=64 time=16.0 ms
64 bytes from 172.16.10.1: icmp_seq=2 ttl=64 time=8.65 ms
64 bytes from 172.16.10.1: icmp_seq=3 ttl=64 time=10.0 ms
^C
--- 172.16.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
```

Figure 4 Ubuntu-2 ping to Ubuntu-1

To identify why pings are successful, the flow table of the OVS can be analysed through the command: `ovs-ofctl dump-flows br0`, what is expected, is a flow where the actions is set to `NORMAL`.

C. *Connect GNS3 Devices to SDN-VM*

The cloud appliance found in the end devices tab on the left sidebar can be used to connect to the local host machine. To configure the cloud to use the correct interface, the following procedure is taken: right-click the cloud, go to configure, tick the `Show special ethernet interfaces` check box, in the drop-down menu select `virbr0`, finally click ok and link the OVS to the new interface. Removal of the Eth0 interface from Br0 is best practice as it is considered the management interface, removal of eth0 can be completed by: `ovs-vsctl del-port br0 eth0`. The same procedure can be taken to configure `eth0` with the appropriate IP as seen in *Figure 3* with its respective IP seen in *Figure 2.* The `ovs-vsctl show | grep eth0` command can be used to verify that `eth0` no longer falls under `br0` as show in *Figure 5.*

```
OpenvSwitch-1
File  Edit  View  Search  Terminal  Help
OpenvSwitch-1:/$ ovs-vsctl show | grep eth0
OpenvSwitch-1:/$ ifconfig eth0
eth0      Link encap:Ethernet  HWaddr D6:1F:1A:C6:38:9D
          inet addr:192.168.122.2  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::d41f:1aff:fec6:389d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Figure 5 Verify OVS eth0 configurations

To verify that virbr0 is on the SDN-VM, opening the local machines terminal and issue: `ifconfig virbr0` to identify that `virbr0` and the IP address is in the same network as OVS-1 eth0's network. This results in the SDN-VM being able to ping both OVS.

D. *Establish Internet Connectivity*

This section uses the NAT cloud to establish internet connectivity and configure OVS-2 `eth0` to use DHCP allocated IP. Editing the file at `/etc/networks/interfaces` and uncommenting the specific lines shown in *Figure 6* and restarting the device will allocate dynamically an IP in the 192.168.122.0/24 network.

```
# Static config for eth0
#auto eth0
#iface eth0 inet static
#       address 192.168.0.2
#       netmask 255.255.255.0
#       gateway 192.168.0.1
#       up echo nameserver 192.168.0.1 > /etc/resolv.conf

# DHCP config for eth0
 auto eth0
 iface eth0 inet dhcp
```

Figure 6 DHCP configuration

```
OpenvSwitch-2:/etc$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=127 time=11.102 ms
64 bytes from 8.8.8.8: seq=1 ttl=127 time=12.070 ms
64 bytes from 8.8.8.8: seq=2 ttl=127 time=12.570 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
```

Figure 7 OVS-2 Pinging Googles public DNS server

E. *Static IP Persistence and DNS Configuration*

The prior interface configurations do not persist restarts and would require manual configuration again. The same `/etc/network/interfaces` file can be edited, however the static IP variables are uncommented.

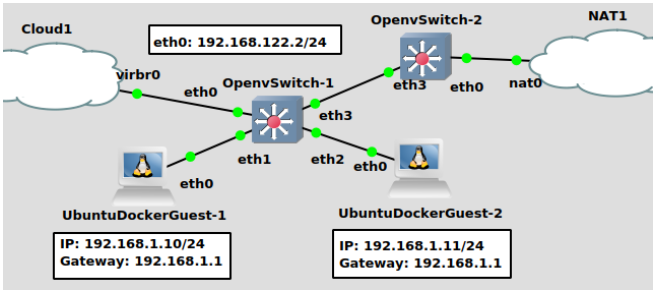

Figure 8 GNS3 Network Topology 2

```
# Static config for eth0
auto eth0
iface eth0 inet static
        address 192.168.1.11
        netmask 255.255.255.0
        gateway 192.168.1.1
#       up echo nameserver 192.168.0.1 > /etc/resolv.conf
```

Figure 9 Static IP configuration to persist restart

The command: `route -n` can be used on an Ubuntu host to confirm default gateway. Note: there is no layer 3 routing capability, therefore, only the two hosts can ping each other, and the SDN-VM can ping OVS-1 `eth0`. To configure the DNS server, edit the `/etc/resolv.conf` file and enter `namespace <IP address>`, Googles public DNS server can be used (8.8.8.8), ensure the changes are saved.

```
OpenvSwitch-2:/$ nslookup 136.186.1.111
Server:         8.8.8.8
Address:        8.8.8.8:53

Non-authoritative answer:
111.1.186.136.in-addr.arpa      name = ns1.swin.edu.au
```

Figure 10 nslookup test using Googles public DNS

V. Discussion

This section provides an informative analysis of major steps in each section of the lab. It explains how the lab questions were answered.

A. Import GNS3 Appliances

The GNS3 appliance marketplace which allows users to import preconfigured devices instead of manual configuration of settings. This benefits in efficiency when testing as you are less likely to encounter issues.

B. GNS3 Appliance Configuration

Switches operating at layer 2 are subject to broadcast storms due to MAC flooding resulting in a loop. STP prevents this. The CPU and RAM usage in GNS3 can be seen in the right-side menu. Configuration of interfaces for each Ubuntu host allows each host to ping each other. IP addressing scheme sets them in the same network, the respective flow which is permitting connectivity can be seen in *Figure 11*.

```
OpenvSwitch-1:/$ ovs-ofctl dump-flows br0
 cookie=0x0, duration=15.119s, table=0, n_packets=22, n_bytes=1604,
 priority=0 actions=NORMAL
OpenvSwitch-1:/$
```

Figure 11 Default flow allowing layer 2 forwarding

As mentioned, the IP addresses are in the same network, therefore layer 2 forwarding. This occurs in hardware does not require an OpenFlow controller to provide instructional flows on handling layer 2 traffic. `actions=NORMAL` indicates layer 2 forwarding, an attempt to ping the SDN-VM or `eth0` on either OVS will fail. `ovs-vsctl show` command will show that every ethernet interface falls under `br0`, therefore frames can be switched from each virtual ethernet interface. Hence why `br0` is used in the STP command as all interfaces are members of it.

C. Connect GNS3 Devices to SDN-VM

As consequence of every interface being a member of `br0`, `eth0` however is described as the management interface in the lab scenario and therefore does not provide meaningful reason to be part of the layer 2 switching network. Therefore, removing it alleviates layer 2 traffic traversing the interface connected to the SDN-VM. The cloud exposes a `virbr0` interface on the local VM and therefore in GNS3 a link can be created from the OVS to `virbr0`. The IP for this virtual bridge is within the 192.168.122.0/24 network; therefore, connectivity is established.

D. Establish Internet Connectivity

The network configuration file found at `/etc/networks/interfaces` provides a way to configure interface properties at startup. *Figure 6* shows the lines to uncomment to establish IP configuration through DHCP at start-up (this is why a restart is required if the configuration is made whilst the device is active), `auto eth0` brings the interface up at start-up, `iface eth0 inet dhcp` tells the device to use IPv4 addressing for this interface. The NAT cloud used runs a DHCP server and runs DNS relay using the default IP network 192.168.122.1/24 [1] (the SDN-VM `virbr0`). A NAT gateway is provided by the NAT cloud and therefore the internet is accessible.

E. Static IP Persistence and DNS Configuration

The similar lines are uncommented in the same configuration file as seen in the previous subsection *Establish Internet Connectivity*. However, the static variables are changed. This results in IP configurations to remain after restarts. The DNS server can be manually configured in the same file by uncommenting the `up echo` line and writing the desired DNS server. This command writes to the designated file the nameserver for which the device should use through static configuration. DNS is required for resolving domain names to IP address or vice versa, and is essential for internet connectivity. Verification of DNS configuration working is utilising `nslookup` as seen in *Figure 10*. Googles DNS server is being used, the DNS request query traverses to 8.8.8.8 and is referred to 136.186.1.111 (Swinburne associated IP), the output is the PTR record which maps IP to a domain name; `ns1.swin.edu.au`.

VI. Conclusion

In conclusion, this lab report demonstrated practical use of GNS3 for creating a virtual network, utilising OpenFlow and OVS technologies. Configuration was demonstrated through means of either static or dynamic and how to have configurations persist device reboots. Host to host and SDN-VM to OVS connectivity was demonstrated to be successful and how the virtual network in GNS3 can be connected to the internet. Various verification commands were shown to demonstrate and reason for why and how configurations were affecting the network to achieve desired results.

VII. References

[1] "Getting Started with GNS3," *GNS3 Documentation*, [Online]. Available: https://docs.gns3.com/docs/ [Accessed: Sep. 27, 2025]

[2] "Appliances," *GNS3 Marketplace*, [Online]. Available: https://www.gns3.com/marketplace/appliances. [Accessed: Sep. 27, 2025]